IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received May 22, 2020, accepted June 11, 2020, date of publication June 16, 2020, date of current version June 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3002761

Acoustic Scene Classification With Squeeze-Excitation Residual Networks

JAVIER NARANJO-ALCAZAR[1,2], (Graduate Student Member, IEEE), SERGI PEREZ-CASTANOS[1], PEDRO ZUCCARELLO[1], AND MAXIMO COBOS[2], (Senior Member, IEEE)

[1] Visualfy, 46181 Benisanó, Spain
[2] Computer Science Department, Universitat de Valencia, 46100 Burjassot, Spain

Corresponding author: Javier Naranjo-Alcazar (javier.naranjo@visualfy.com)

This work was supported by the European Union's Horizon 2020 Research and Innovation Program under Grant 779158. The work of Javier Naranjo-Alcazar was supported by the Spanish Ministry of Science, Innovation, and Universities, through the Torres Quevedo Fellowship, under Grant DIN2018-009982. The work of Pedro Zuccarello was supported by the Spanish Ministry of Science, Innovation, and Universities, through the Torres Quevedo Fellowship, under Grant PTQ-17-09106. The work of Maximo Cobos was supported in part by the European Regional Development Fund (ERDF), and in part by the Spanish Ministry of Science, Innovation, and Universities under Grant RTI2018-097045-B-C21.

ABSTRACT Acoustic scene classification (ASC) is a problem related to the field of machine listening whose objective is to classify/tag an audio clip in a predefined label describing a scene location (e. g. park, airport, etc.). Many state-of-the-art solutions to ASC incorporate data augmentation techniques and model ensembles. However, considerable improvements can also be achieved only by modifying the architecture of convolutional neural networks (CNNs). In this work we propose two novel squeeze-excitation blocks to improve the accuracy of a CNN-based ASC framework based on residual learning. The main idea of squeeze-excitation blocks is to learn spatial and channel-wise feature maps independently instead of jointly as standard CNNs do. This is usually achieved by combining some global grouping operators, linear operators and a final calibration between the input of the block and its learned relationships. The behavior of the block that implements such operators and, therefore, the entire neural network, can be modified depending on the input to the block, the established residual configurations and the selected non-linear activations. The analysis has been carried out using the TAU Urban Acoustic Scenes 2019 dataset presented in the 2019 edition of the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge. All configurations discussed in this document exceed the performance of the baseline proposed by the DCASE organization by 13% percentage points. In turn, the novel configurations proposed in this paper outperform the residual configurations proposed in previous works.

INDEX TERMS Acoustic scene classification, deep learning, machine listening, pattern recognition, squeeze-excitation.

I. INTRODUCTION

The analysis of everyday ambient sounds can be very useful when developing intelligent systems in applications such as domestic assistants, surveillance systems or autonomous driving. Acoustic scene classification (ASC) is one of the most typical problems related to machine listening [1]–[4]. Machine listening is understood as the field of artificial intelligence that attempts to create intelligent algorithms capable of extracting meaningful information from audio data. Therefore, ASC can be defined as the area of machine listening that attempts to tag an audio clip in one of the predefined tags related to the description of a scene (for example, airport, park, subway, etc.).

The first approaches to the ASC problem were centered on the design of proper inputs to the classifier, this is, feature engineering [5]. Most research efforts tried to create meaningful representations of the audio data to later feed gaussian mixture models (GMMs), hidden Markov models (HMM) or support vector machines (SVMs) [6]. In this context, a wide range of input representations were proposed such as Mel-frequency cepstral coefficients (MFCCs) [7], [8],

The associate editor coordinating the review of this manuscript and approving it for publication was Shiqing Zhang.

Wavelets [8], constant-Q transform (CQT) or histograms of oriented gradientes (HOG) [9], among others.

With the years and the emergence of convolutional networks in the field of image and computer vision, CNNs have become a preferred option for the design of machine listening systems, usually fed with a 2D audio representation such as log-Mel spectrograms [1], [10]. These networks have shown very satisfactory results, especially when they are trained on large datasets. This is why data augmentation techniques are commonly applied, such as mixup strategies [11] or temporal cropping [12]. In addition, to improve the final accuracy, many studies use ensembles, combining the output from different classifiers to obtain a single more robust prediction. Unfortunately, the use of ensembles makes it more difficult to analyze the contribution to the classification performance of a new CNN architecture integrated within the proposed ensemble. To avoid such issue, this work considers isolated contributions of several CNN architectures implemented with different residual blocks based on squeeze-excitation, without any extra modifications during the training or inference phases.

CNNs are built with stacked convolutional layers. These layers learn its filter coefficients by capturing local spatial relationships (neighbourhood information) along the input channels and generate features maps (filtered inputs) by jointly encoding the spatial and channel information. In all application domains (image classification/segmentation, audio classification/tagging, etc.), the idea of encoding the spatial and the channel information independently has been less studied, despite having shown promising results [13], [14].

In order to provide insight about the behaviour of CNNs when analyzing spatial and channel information independently, several squeeze-excitation (SE) blocks have been presented in the image classification literature [13], [14]. In [14], a block that ''squeezes'' spatially and ''excites'' channel-wise with linear relationships was presented. The idea behind this block, denoted as *cSE* in this work, is to model the interdependencies between the channels of feature maps by exciting in a channel-wise manner. This type of block showed its effectiveness in image classification tasks, outperforming other state-of-the-art networks only by inserting it at a specific point of the network. Following this idea, two more blocks were presented in [13]. The first one, denoted as *sSE*, ''squeezes'' along the channels and ''excites'' spatially, whereas the last block, *scSE*, combines both strategies. The scSE block recalibrates the feature maps along spatial and channel dimensions independently (*cSE* and *sSE*) and then combines the information of both paths by adding their outputs. This last block showed the most promising results in image-related tasks. According to [13], this block forces the feature maps to be more informative, both spatially and channel-wise.

This work analyzes the performance of conventional SE blocks for addressing the ASC problem and proposes two novel block configurations in this context. The new configurations are intended to enhance the benefits of residual learning and feature map recalibration in a jointly fashion. This is achieved by a double short-cut connection that enforces residual learning both with and without recalibrated outputs. The use of SE techniques allows the network to extract more meaningful information during training, while residual learning facilitates the training procedure by mitigating vanishing gradient problems. The results show that, by using the proposed block configurations, results are considerably improved. Moreover, it is shown that all the residual SE configurations perform better than a classical convolutional residual block in the considered task.

The following of the paper is organized as follows. Section II presents the the background for the techniques used in this work in the context of ASC, namely Squeeze-Excitation and residual learning. Section III introduces the different SE blocks analyzed in this work and the baseline CNN architecture. Section IV describes the dataset used in the experiments, the audio pre-processing and the training procedure of the CNN. Section V discusses the experimental results, while Section VI concludes our work.

II. BACKGROUND

This section summarizes the technical background for this work and describes the ideas underlying SE blocks and residual networks.

A. RELATED WORK

Some previous works have shown that the use of SE modules can be a simple and effective approach to tackle audio classification problems. In [15], a multi-scale fusion and channel weighted CNN was proposed within an ASC context. The framework consists of two stages: a multi-scale feature fusion scheme that integrates a hierarchy of semantic-features extracted from a simplified Xception architecture, and a final SE-based channel weighting stage. However, such work considers only channel recalibration by using a *cSE*-like block at a final stage, without further integration of other SE-based calibration modules. In contrast, the configurations proposed in this work consider both spatial and channel-wise weighting within a residual learning framework jointly and at multiple depths within the network architecture.

Another work using SE techniques in the audio domain is [16], which presented a VGG-style CNN and compared its performance with an enhanced version including residual connections and SE modules. In contrast to the work presented in this paper, an end-to-end 1D architecture accepting raw audio inputs was proposed, with *cSE* channel-wise recalibration. The results over three different tasks (music auto-tagging, speech command recognition and acoustic event detection) confirmed the superiority of the enhanced network.

Finally, although some technical reports could not corroborate the improvements offered by SE modules over plain residual networks in audio-oriented tasks [17], few details were given, which motivates further the analysis carried out in this work.

B. SQUEEZE-EXCITATION BLOCKS

Squeeze-excitation (SE) blocks can be understood as modules for channel recalibration of feature maps [13]. Let' s assume an input feature map, $\mathbf{X} \in \mathbb{R}^{H\times W\times C'}$, that feeds any convolutional block, usually implemented by convolutional layers and non-linearities, and generates an output feature map $\mathbf{U} \in \mathbb{R}^{H\times W\times C}$. Here, $\mathbf{U}$ could also be expressed as $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_C]$, being $\mathbf{u}_i \in \mathbb{R}^{H\times W}$ a channel output. Considering this notation, H and W represents the height and the width, while C' and C defines the number of input and output channels, respectively. The convolutional process function can be defined as $\mathbf{F}(\cdot)$, so that $\mathbf{F}(\mathbf{X}) = \mathbf{U}$. The output $\mathbf{U}$ is generated by combining the spatial and channel information of $\mathbf{X}$. The objective of SE blocks is to recalibrate $\mathbf{U}$ with $\mathbf{F}_{SE}(\cdot)$ to generate $\hat{\mathbf{U}}$, i.e. $\mathbf{F}_{SE}(\cdot) : \mathbf{U} \rightarrow \hat{U}$. This recalibrated feature map, $\hat{\mathbf{U}}$, can be stacked after every convolutional block and then used as input to the forthcoming pooling layers. This recalibration can be carried out with different types of block functions $\mathbf{F}_{SE}(\cdot)$, as it is next explained.

1) SPATIAL SQUEEZE AND CHANNEL EXCITATION BLOCK (cSE)

In a *cSE* module (depicted in Fig. 1(a)) for spatial squeeze and channel excitation, a unique feature map of each channel from $\mathbf{U}$ is first obtained by means of global average pooling. This operator produces a vector $\mathbf{z} \in \mathbb{R}^{1\times 1\times C}$. The kth element of such vector can be expressed as:

$$z_k = \frac{1}{H \times W}\sum_{i}^{H}\sum_{j}^{W}\mathbf{u}_k(i,j), \quad k = 1, \dots, C, \qquad (1)$$

where $\mathbf{u}_k(i,j)$ denotes the (i,j) element of the kth channel feature map.

As suggested by Eq. (1), global spatial information is embedded in vector $\mathbf{z}$. This representation is then used to extract channel-wise dependencies using two fully-connected layers, obtaining the transformed vector $\hat{\mathbf{z}}$. Therefore, $\hat{\mathbf{z}}$ can be expressed as $\hat{\mathbf{z}} = \mathbf{W}_1(\delta(\mathbf{W}_2\mathbf{z}))$, where δ represents ReLU activation. $\mathbf{W}_1 \in \mathbb{R}^{C\times\frac{C}{\rho}}$ and $\mathbf{W}_2 \in \mathbb{R}^{\frac{C}{\rho}\times C}$ are the weights of the fully-connected layers, and ρ is a ratio parameter. As last step, the activation range is compressed to the interval [0, 1] using a sigmoid activation function, σ. This final step indicates the importance of each channel and how they should be rescaled. The purpose of this recalibration is to let the network ignore channels with less information and emphasize the ones that provide more meaningful information. Then, the rescaled feature maps, $\hat{\mathbf{U}}$, can be expressed as [13], [14]:

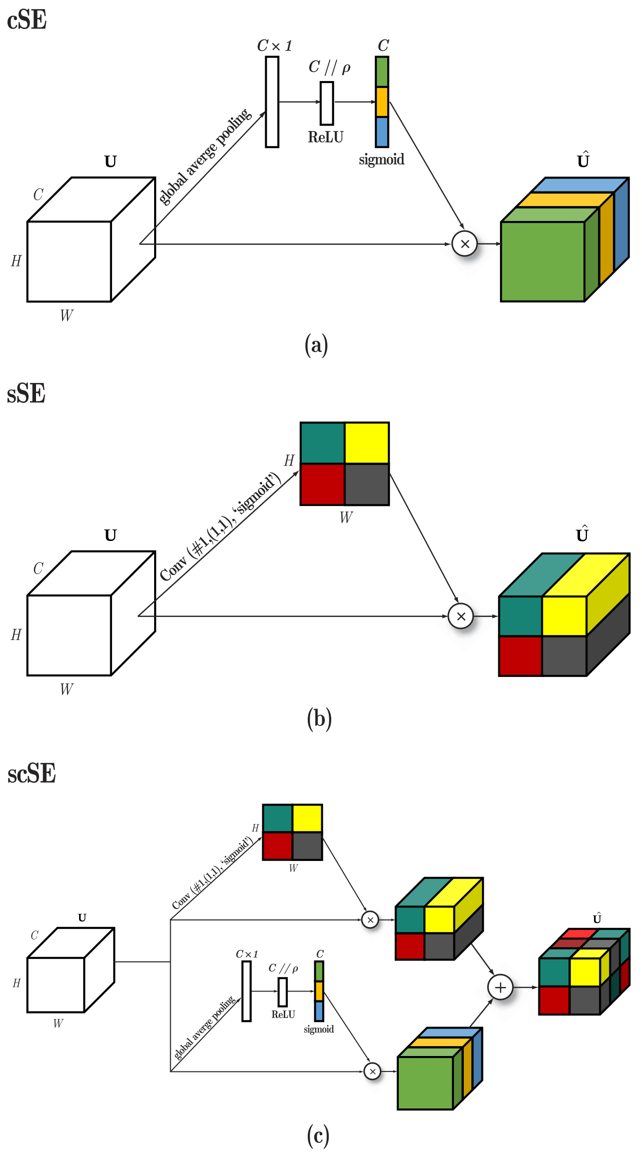
$$\hat{\mathbf{U}}_{cSE} = \mathbf{F}_{cSE}(\mathbf{U}) = [\sigma(\hat{z}_1)\mathbf{u}_1, \dots, \sigma(\hat{z}_C)\mathbf{u}_C], \qquad (2)$$

where $\hat{z}_k$ are the elements of the transformed vector $\hat{\mathbf{z}}$.

2) CHANNEL SQUEEZE AND SPATIAL EXCITATION BLOCK (sSE)

In the case of an *sSE* block [13], as shown in Fig. 1(b), a unique convolutional layer with one filter and (1, 1) kernel size is implemented to obtain a channel squeeze and spatial excitation effect. Here, it is assumed an alternative representation of the input tensor as $\mathbf{U} = [\mathbf{u}^{1,1}, \mathbf{u}^{1,2}, \dots, \mathbf{u}^{i,j}, \dots, \mathbf{u}^{H,W}]$ where $\mathbf{u}^{i,j} \in \mathbb{R}^{1\times 1\times C}$. The convolution can be expressed as $\mathbf{q} = \mathbf{W} \star \mathbf{U}$, being $W \in \mathbb{R}^{1\times 1\times C\times 1}$ and $q \in \mathbb{R}^{H\times W}$. Each $q_{i,j}$ represents the combination of all channels in location (i,j). As done with *cSE*, the output of this convolution is passed through a sigmoid function. Each $\sigma(q_{i,j})$ determines the importance of the specific location (i,j) across the feature map. Like the previous block, this recalibration process indicates which locations are more meaningful during the training procedure. As a result, the output of the SE block can be expressed as [13]:

$$\hat{\mathbf{U}}_{sSE} = \mathbf{F}_{sSE}(U) = [\sigma(q_{1,1})u^{1,1}, \dots, \sigma(q_{H,W})u^{H,W}]. \qquad (3)$$



(a)

(b)

(c)

FIGURE 1. Diagram of different SE blocks: (a) describes cSE block procedure, (b) ilustrates sSE block framework and (c) shows scSE block by combining (a) and (b).

3) SPATIAL AND CHANNEL SQUEEZE & EXCITATION BLOCK (scSE)

The *scSE* block [13] is implemented by declaring *cSE* and *sSE* blocks in parallel and adding both outputs (see Fig. 1(c)). It has been reported that the *scSE* block shows better performance than *cSE* and *sSE* used independently. In this case, a location (i, j, c) gets a higher sigmoid or activation value when both channel and spatial recalibration get it at the same time [13]:

$$\hat{\mathbf{U}}_{scSE} = \hat{\mathbf{U}}_{cSE} + \hat{\mathbf{U}}_{sSE}. \tag{4}$$

In this case, the network focuses on feature maps that are meaningful from both a spatial and channel-wise point of view.

C. RESIDUAL NETWORKS

Residual networks were first proposed in [18]. A network of this kind replaces the standard stacked convolutional layers [19] by residual blocks. Residual layers are designed to approximate a residual function: $\mathcal{F}(\mathbf{X}) := \mathcal{H}(\mathbf{X}) - \mathbf{X}$, where $\mathcal{H}(\cdot)$ represents the mapping to be fit by a set of stacked layers and $\mathbf{X}$ represents the input to the first of such stacked layers. The original function $\mathcal{H}$ can therefore be defined as $\mathcal{H}(\mathbf{X}) = \mathcal{F}(\mathbf{X}) + \mathbf{X}$. The main motivation of choosing this kind of network corresponds to the intuition that optimizing a residual mapping may be easier than optimizng the original unreferenced one, as in a classical convolutional network. A simple way of implementing residual learning in CNNs is by adding a shortcut connection that performs as an identity mapping, adding back the input $\mathbf{X}$ to the output of the residual block $\mathcal{F}(\mathbf{X})$. In the first proposition of the residual block, Rectified ReLU activation is applied after the addition and the result of such activation becomes the input for the next residual block. Note, that in the first configuration, shortcut connections do not add more parameters nor extra computational cost. Therefore, deeper networks can be trained with little additional effort, reducing vanishing-gradient problems. As it will be later explained, in this work, the identity mapping is replaced with a 1 × 1 convolutional layer as it is explained in Section III. Therefore, this work function can be expressed as $\mathcal{H}(\mathbf{X}) = \mathcal{F}(\mathbf{X}) + g(\mathbf{X})$, where $g(\cdot)$ represents the convolutional process with the learnt filter coefficients.

III. CONFIGURATIONS FOR SQUEEZE-AND-EXCITATION RESIDUAL NETWORKS

According to [14], SE blocks exhibit better performance when deployed on networks with residual configuration than on VGG-style networks. Therefore, two novel residual blocks implementing *scSE* modules are presented in this paper. The performance of these two newly proposed blocks is compared against other state-of-the-art residual configurations that incorporate SE modules.

A. SE BLOCK DESCRIPTION

All the configurations analyzed in this work are depicted in Fig. 2. In the following, we describe in details these blocks.

1) Conv-RESIDUAL

Shown in Fig. 2(a), is inspired by [18]. It is used as a baseline in order to validate the network performance without any SE and how much it can be improved when incorporating these blocks. In the present work some slight modifications for a more convenient implementation were introduced: the shortcut connection was implemented with a 1 × 1 convolutional layer and the activation after the addition was set to an exponential linear unit (ELU) function [20], [21].

2) Conv-POST

Shown in Fig. 2(b), is inspired by the block referred to as *se-POST* in [14]. The *scSE* block is included at the end and is equivalent to a recalibration of the *Conv-residual* block.

3) Conv-POST-ELU

Shown in Fig. 2(c), is very similar to the above *Conv-POST* block, but the recalibration is performed over the ELU-activated output of the residual block.

4) Conv-STANDARD

Shown in Fig. 2(d), is inspired by [14], where the *scSE* block is stacked after the convolutional block for recalibrating prior to adding the shortcut branch.

5) Conv-StandardPOST

Shown in Fig. 2(e) is proposed in this work to create a double shortcut connection, one before SE calibration and one after. The idea is to let the network learn residual mappings simultaneously with and without SE recalibration, thus, affecting the way in which the block optimizes the residual by considering jointly standard and post SE-calibrated outputs.

6) Conv-StandardPOST-ELU

Shown in Fig. 2(f) is the other proposed block, corresponding to the above explained *Conv-StandardPOST* block, but followed by ELU activation.

To summarize, the output $\mathbf{X}_{l+1}$ of each block for an input $\mathbf{X}$ is given by:

$$\text{a)}\quad \mathbf{X}_{l+1} = \mathcal{R}\left(\mathcal{F}(\mathbf{X}) + g(\mathbf{X})\right), \tag{5}$$

$$\text{b)}\quad \mathbf{X}_{l+1} = \mathbf{F}_{SE}\left(\mathcal{F}(\mathbf{X}) + g(\mathbf{X})\right), \tag{6}$$

$$\text{c)}\quad \mathbf{X}_{l+1} = \mathbf{F}_{SE}\left(\mathcal{R}\left(\mathcal{F}(\mathbf{X}) + g(\mathbf{X})\right)\right), \tag{7}$$

$$\text{d)}\quad \mathbf{X}_{l+1} = \mathcal{F}_{(SE)}(\mathbf{X}) + g(\mathbf{X}), \tag{8}$$

$$\text{e)}\quad \mathbf{X}_{l+1} = \mathbf{F}_{SE}\left(\mathcal{R}\left(\mathcal{F}(\mathbf{X}) + g(\mathbf{X})\right)\right) + g(\mathbf{X}), \tag{9}$$

$$\text{f)}\quad \mathbf{X}_{l+1} = \mathcal{R}\left(\mathbf{F}_{SE}\left(\mathcal{R}\left(\mathcal{F}(\mathbf{X}) + g(\mathbf{X})\right)\right) + g(\mathbf{X})\right), \tag{10}$$

where $\mathcal{R}(\cdot)$ refers to ELU activation function with α parameter set to 1 and $\mathcal{F}_{(SE)}$ denotes a residual function that includes SE calibration. As it will be discussed in Section V, the two proposed configurations have been shown to outperform the rest in the considered acoustic scene analysis task.

In order to avoid possible duplications or expansion processes in the channel dimension, the identity branch is replaced by a convolutional layer with a $(1, 1)$ kernel size

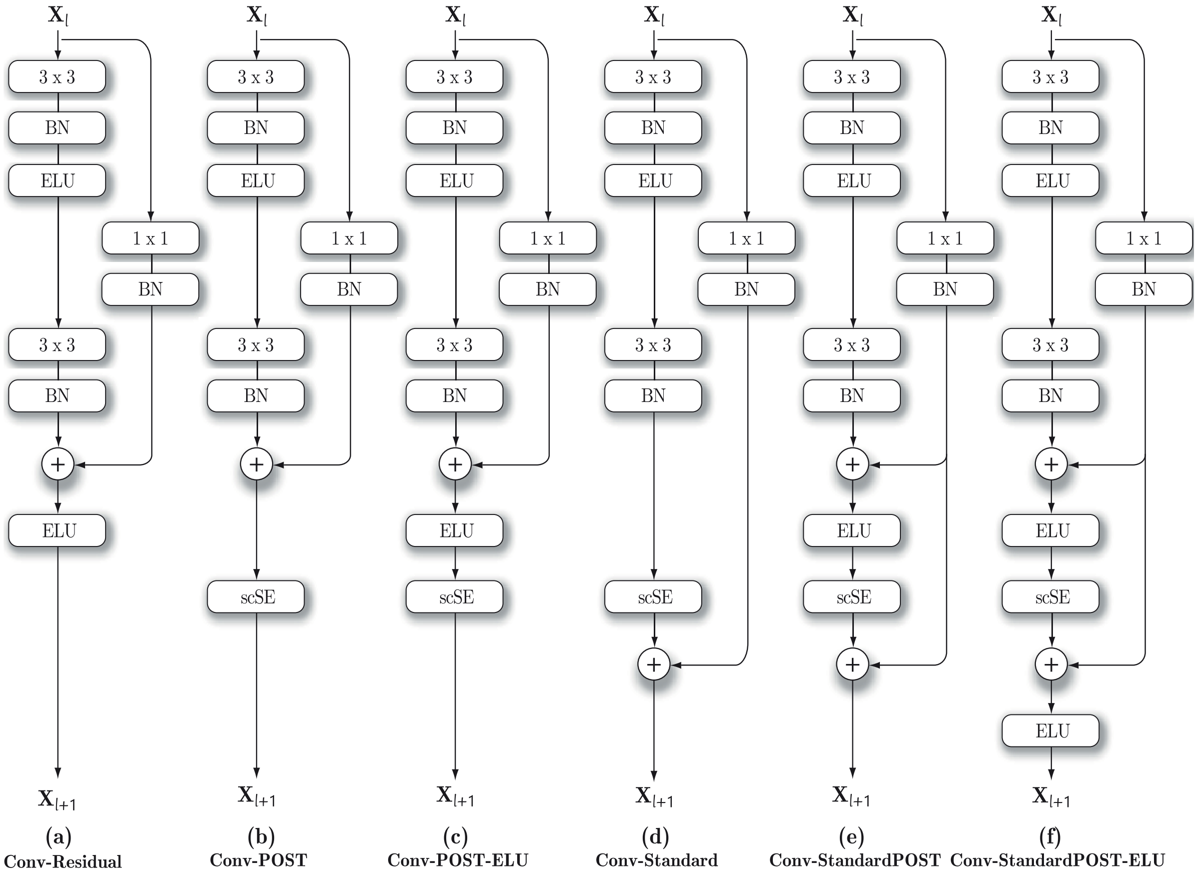


FIGURE 2. **Different residual squeeze-excitation blocks analyzed in this work: (a) is inspired by the first residual block proposed in [18]; (b), (c) and (d) are inspired by the work done in [14]; (e) and (f) are the two novel configurations proposed in this work.**

and with the same number of filters as the residual branch. Including such convolutional layer in the shortcut branch creates a projection that avoids dimensionality conflicts in the residual block addition.

By looking at Fig. 2, it can be clearly observed that the most representative feature of the two proposed blocks, (e) and (f), resides in the use of two skip connections: one before SE re-calibration and one after. This double short-cut connection leads the network towards the learning of a global residual function embedding an inner and SE-calibrated partial residual. The objective is to facilitate the learning of calibration weightings by using the same residual rationale.

In general, the presence or absence of relevant acoustic events within an input audio clip can be very important when addressing the ASC problem. The use of spatial and channel-wise recalibration at different depths of the network adds a mechanism to allow the network weight properly, according to their importance, the different dimensions of the information flowing throughout the network. Therefore, SE modules are expected to add flexibility for identifying relevant acoustic textures or events, making easier to infer the type of underlying acoustic scene.

B. NETWORK ARCHITECTURE

The CNN implemented in order to validate the behaviour of the different SE configurations has been inspired on [22] where a VGG-style [19] network with 3 convolutional blocks followed by different max-pooling and dropout [23] operators is implemented. In the present work, the original convolutional blocks have been replaced with the different residual squeeze-excitation blocks proposed in this study. The max-pooling, dropouts and linear layers are configured with the same parameters as in [22]. The network architecture can be found in Table 1.

As the database used in the current work is much smaller than the one in [14], some of the hyperparameters that define the components of the scSE block had to be modified. The number of elements in the Dense layer with ReLU activation in Fig. 1(a) has been set to 16 in the first Residual-scSE block, the same as in [14] in its cSE block, but the number of filters at the input, C, has been set to $C = 32$. Therefore, the ratio between these parameters throughout the network is two, as observed Table 1. The number of network parameters that implement SE residual blocks, i.e. those represented in Fig. 2(b)-(f), is 528,334. On the other hand, the network that does not integrate SE modules has 506,606. Note, therefore, that there is only a slight increase of approximately 4% in the

TABLE 1. Proposed network for validating the scSE configurations of Fig. 2. Values preceeded by # correspond to the number of filters. Kernel sizes are set as indicated in Fig. 2. This architecture is inspired by the work in [22].

Residual-scSE block (#32, $\rho = 2$)
MaxPooling(2,10)
Dropout(0.3)
Residual-scSE block (#64, $\rho = 2$)
MaxPooling(2,5)
Dropout(0.3)
Residual-scSE block (#128, $\rho = 2$)
MaxPooling(2,5)
Dropout(0.3)
Flatten
[Dense(100), batch normalization, ELU]
Dropout(0.4)
[Dense(10), batch normalization, softmax]

SE networks. Table 2 shows as well the number of floating point operations (FLOPs) involved in each network.

IV. EXPERIMENTAL DETAILS

This section describes in detail the experimental implementation carried out to conduct the analysis of the presented SE residual blocks, including the datasets, the audio representation selected to feed the network and the training configuration.

A. DATASET

To check the behavior of these implementations in an ASC problem, the TAU Urban Acoustic Scenes 2019, Development dataset presented in Task 1A of the 2019 edition of DCASE has been used [10]. The database consists of 40 hours of stereo audio-recording in different urban environments and landscapes such as parks, metro stations, airports, etc. making a total of 10 different scenes. These have been recorded in different cities such as Barcelona, Paris or Helsinki, among others. All audio clips are 10-second long. They are divided into two subsets of 9185 and 4185 clips for training and validation, respectively. Although there are a slightly different number of samples available for each class, the data set is not severely unbalanced.

B. AUDIO PROCESSING

The input to the network is a 2D log-Mel spectrogram representation with 3 audio channels. The three channels are composed of the harmonic and percussive component [24], [25] of the signal converted to mono and the difference between left (L) and right (R) channels. That is, the first channel corresponds to the log-Mel spectrogram of the harmonic source, the second channel corresponds to the same representation but over the percussive source and the last one to the log-Mel spectrogram of the difference between channels calculated by subtracting left and right channels ($L - R$). This representation, known as HPD, was presented in [22]. The log-Mel spectrogram is calculated using 64 Mel filters with a window size of 40 ms and 50% overlap. Therefore, an audio clip becomes a $64 \times T \times 3$ array with T being the number of time frames. In this specific dataset, the input audio representation corresponds to an array of dimension $64 \times 500 \times 3$.

TABLE 2. Parameters and FLOPs analysis from the studied network configurations.

Residual block	Network parameters	Network FLOPs
Conv-Residual	506606	1009115
Conv-POST Conv-POST-ELU Conv-Standard Conv-StandardPOST Conv-StandardPOST-ELU	528334	1052571

C. TRAINING PROCEDURE

The training process was optimized using the Adam optimizer [26]. The cost function used was the categorical crossentropy. Training was limited to a maximum of 500 epochs but early stopping is applied if the validation accuracy does not improve by 50 epochs. If this same metric does not improve in 20 epochs, the learning rate is decreased by a factor of 0.5. The batch size used was 32 samples.

V. RESULTS

In order to analyze the contributions of this work with respect to other state-of-the-art approaches, the results obtained with the different configurations presented in this work (see Fig. 2) are compared to the ones obtained by different authors in Task 1A of DCASE 2019 using the same dataset. For a fair comparison, only submissions not making use of data augmentation techniques are considered. In the case of submissions that presented an ensemble of several models, only the results of the best performing model making up the ensemble are taken into account. For example, in [27] a global development accuracy of 78.3% is reported, but that value was obtained by averaging 5 models. The best individual model obtained 72.4%, so this is the value presented in Table 3. This said, please be aware that the accuracy of the final submission[1] may differ from that presented in Table 3. Next, we summarize some important features of the competing approaches.

- **Wang_NWPU_task1a** [27]: the audio representation considers two channels using a log-Mel Spectrogram from harmonic and percussive sources similar to our representation. The number of Mel filters is set to 256. The window size is set to 64 ms and the hop size to 15 ms. Mel filters are calculated with cutoff frequencies from 50 Hz to 14 kHz. A VGG-style CNN [19] is used as a classifier.

[1] http://dcase.community/challenge2019/task-acoustic-scene-classification-results-a

TABLE 3. Accuracy results from the validation partition in development phase.

System	Development accuracy (%)
Baseline [10]	62.5
Wang_NWPU_task1a [27]	72.4
Fmta91_KNToosi_task1a [28]	70.49
MaLiu_BIT_task1a [29]	76.1 (evaluation)
DSPLAB_TJU_task1a [30]	64.3
Kong_SURREY_task1a [31]	69.2
Liang_HUST_task1a [32]	70.70
Salvati_DMIF_task1a [33]	69.7
Conv-Residual	*74.51 ± 0.65*
Conv-Standard	*75.16 ± 0.33*
Conv-POST	*75.84±0.65*
Conv-POST-ELU	*75.81±0.47*
Conv-StandardPOST	***76.72±0.59***
Conv-StandardPOST-ELU	*76.00±0.55*

- **Fmta91_KNToosi_task1a** [28]: wavelet scattering spectral features are extracted from the mono audio signal. A random subspace method is used as classifier.
- **MaLiu_BIT_task1a** [29]: Deep Scattering Spectra features (DSS) are extracted from each stereo channel. Classification is performed with a Convolutional Recurrent Neural Network (CRNN). For this network, Table 3 does not report the accuracy on the development set (only on the evaluation set). This is because of some mismatch reported by the authors in the validation procedure with the configuration of the dataset.
- **DSPLAB_TJU_task1a** [30]: this submission approaches the problem in a more classical way extracting audio statistical features such as ZRC, RMSE, spectrogram centroid, etc. A GMM is used as a classifier.
- **Kong_SURREY_task1a** [31]: this submssion can be defined as the state-of-the-art framework in ASC problem. The audio representation considers also the log-Mel spectrogram. The classifier is a VGG-based [19] CNN. This network is a fully convolutional network with no linear layers implemented. The feature maps are reshaped into a one dimensional vector using a global average pooling before the decision layer.
- **Liang_HUST_task1a** [32]: in this method, the log-Mel spectrogram is first extracted after converting the audio signal to mono. Interestingly, the log-Mel spectrogram is divided into two-seconds spectrograms, that means that spectrogram shapes change from $[F \times T \times 1]$ to $[F \times (T/5) \times 1]$. This configuration allows training with audio samples consisting of 5 different spectrograms instead of one. A CNN with frequency attention mechanism is implemented as classifier. For more detail of the attention implementation, see [32].
- **Salvati_DMIF_task1a** [33]: unlike the other submissions, this one works directly on the audio vector. To this end, a 1D convolutional network is implemented. Although some recent efforts have been made in this direction [34], the state-of-the-art literature shows that 2D audio representations, such as spectrograms, still obtain the better classification results [35].
- **DCASE baseline** [10]: the audio is first converted to mono and a log-Mel spectrogram is extracted. In this case, only 40 Mel bins are calculated instead of 64, which is the typical state-of-the-art choice. A CNN is used as a classifier with 2 convolutional layers. The 1D conversion before classification layers is performed by a flatten layer. A dense layer is stacked before the decision layer.

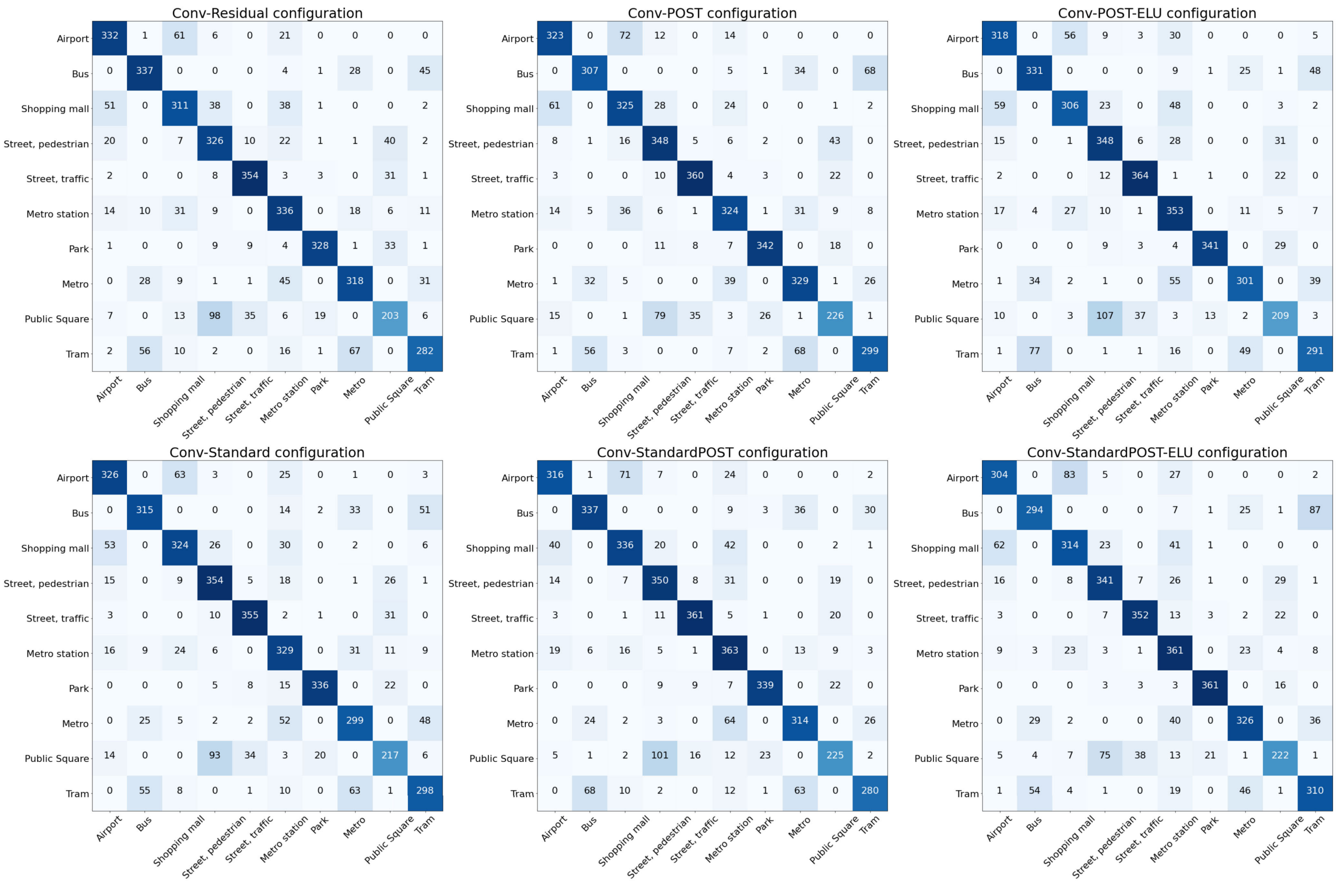
A. GLOBAL PERFORMANCE

Although the results of the DCASE challenge only report the mean accuracy value, we consider 10 runs to provide not only the mean accuracy value, but also the standard deviation. As it can be seen in Table 3, all the configurations detailed in Fig. 2 obtain better accuracy than the DCASE baseline. The contribution of the scSE block is easily justified as *Conv-Residual* gets the lowest performance among the studied configurations. In general, *POST* configurations show a slight improvement compared to the *Standard* configuration. This behaviour differs from what was reported in the original paper, [14], in which these blocks were analyzed in the image domain, where the Standard block outperforms the POST block. There is no remarkable difference between *Conv-POST* and *Conv-POST-ELU*. It is also shown that the networks that incorporate the two novel blocks presented in this work, the ones depicted in Figs. 2(e) and (f), exhibit the best accuracy values. The shortcut addition at two differente points of the residual block, this is, before and after the scSE block, allowed the network to obtain a more precise classification in this ASC task.

B. CLASS-WISE PERFORMANCE

Fig. 3 shows confusion matrices for each of the analyzed residual blocks in this work. In general, the performance across the different classes is considerably balanced. The "Public square" class is the one showing the worst performance, tending to be misclassified as "Street, Pedestrian". Other similar classes such as "Airport" and "Shopping mall" or "Tram" and "Bus" or "Metro" tend also to produce common errors in the analyzed networks.

By analyzing the class-wise performance of the two proposed blocks with respect to the conventional *Conv-Residual* block, substantial improvements are observed. Considering the proposed *Conv-StandardPOST* block, a significant improvement is observed for the classes "Metro station" and "Street, Pedestrian". Other classes showing slight improvements are "Shopping mall", "Park" or "Public square". The class showing the worst relative result was "Airport". On the other hand, the second proposed block *Conv-StandardPOST-ELU* provides substantial improvements in "Street traffic"



Conv-Residual configuration

	Airport	Bus	Shopping mall	Street, pedestrian	Street, traffic	Metro station	Park	Metro	Public Square	Tram
Airport	332	1	61	6	0	21	0	0	0	0
Bus	0	337	0	0	0	4	1	28	0	45
Shopping mall	51	0	311	38	0	38	1	0	0	2
Street, pedestrian	20	0	7	326	10	22	1	1	40	2
Street, traffic	2	0	0	8	354	3	3	0	31	1
Metro station	14	10	31	9	0	336	0	18	6	11
Park	1	0	0	9	9	4	328	1	33	1
Metro	0	28	9	1	1	45	0	318	0	31
Public Square	7	0	13	98	35	6	19	0	203	6
Tram	2	56	10	2	0	16	1	67	0	282

Conv-POST configuration

	Airport	Bus	Shopping mall	Street, pedestrian	Street, traffic	Metro station	Park	Metro	Public Square	Tram
Airport	323	0	72	12	0	14	0	0	0	0
Bus	0	307	0	0	0	5	1	34	0	68
Shopping mall	61	0	325	28	0	24	0	0	1	2
Street, pedestrian	8	1	16	348	5	6	2	0	43	0
Street, traffic	3	0	0	10	360	4	3	0	22	0
Metro station	14	5	36	6	1	324	1	31	9	8
Park	0	0	0	11	8	7	342	0	18	0
Metro	1	32	5	0	0	39	0	329	1	26
Public Square	15	0	1	79	35	3	26	1	226	1
Tram	1	56	3	0	0	7	2	68	0	299

Conv-POST-ELU configuration

	Airport	Bus	Shopping mall	Street, pedestrian	Street, traffic	Metro station	Park	Metro	Public Square	Tram
Airport	318	0	56	9	3	30	0	0	0	5
Bus	0	331	0	0	0	9	1	25	1	48
Shopping mall	59	0	306	23	0	48	0	0	3	2
Street, pedestrian	15	0	1	348	6	28	0	0	31	0
Street, traffic	2	0	0	12	364	1	1	0	22	0
Metro station	17	4	27	10	1	353	0	11	5	7
Park	0	0	0	9	3	4	341	0	29	0
Metro	1	34	2	1	0	55	0	301	0	39
Public Square	10	0	3	107	37	3	13	2	209	3
Tram	1	77	0	1	1	16	0	49	0	291

Conv-Standard configuration

	Airport	Bus	Shopping mall	Street, pedestrian	Street, traffic	Metro station	Park	Metro	Public Square	Tram
Airport	326	0	63	3	0	25	0	1	0	3
Bus	0	315	0	0	0	14	2	33	0	51
Shopping mall	53	0	324	26	0	30	0	2	0	6
Street, pedestrian	15	0	9	354	5	18	0	1	26	1
Street, traffic	3	0	0	10	355	2	1	0	31	0
Metro station	16	9	24	6	0	329	0	31	11	9
Park	0	0	0	5	8	15	336	0	22	0
Metro	0	25	5	2	2	52	0	299	0	48
Public Square	14	0	0	93	34	3	20	0	217	6
Tram	0	55	8	0	1	10	0	63	1	298

Conv-StandardPOST configuration

	Airport	Bus	Shopping mall	Street, pedestrian	Street, traffic	Metro station	Park	Metro	Public Square	Tram
Airport	316	1	71	7	0	24	0	0	0	2
Bus	0	337	0	0	0	9	3	36	0	30
Shopping mall	40	0	336	20	0	42	0	0	2	1
Street, pedestrian	14	0	7	350	8	31	0	0	19	0
Street, traffic	3	0	1	11	361	5	1	0	20	0
Metro station	19	6	16	5	1	363	0	13	9	3
Park	0	0	0	9	9	7	339	0	22	0
Metro	0	24	2	3	0	64	0	314	0	26
Public Square	5	1	2	101	16	12	23	0	225	2
Tram	0	68	10	2	0	12	1	63	0	280

Conv-StandardPOST-ELU configuration

	Airport	Bus	Shopping mall	Street, pedestrian	Street, traffic	Metro station	Park	Metro	Public Square	Tram
Airport	304	0	83	5	0	27	0	0	0	2
Bus	0	294	0	0	0	7	1	25	1	87
Shopping mall	62	0	314	23	0	41	1	0	0	0
Street, pedestrian	16	0	8	341	7	26	1	0	29	1
Street, traffic	3	0	0	7	352	13	3	2	22	0
Metro station	9	3	23	3	1	361	0	23	4	8
Park	0	0	0	3	3	3	361	0	16	0
Metro	0	29	2	0	0	40	0	326	0	36
Public Square	5	4	7	75	38	13	21	1	222	1
Tram	1	54	4	1	0	19	0	46	1	310

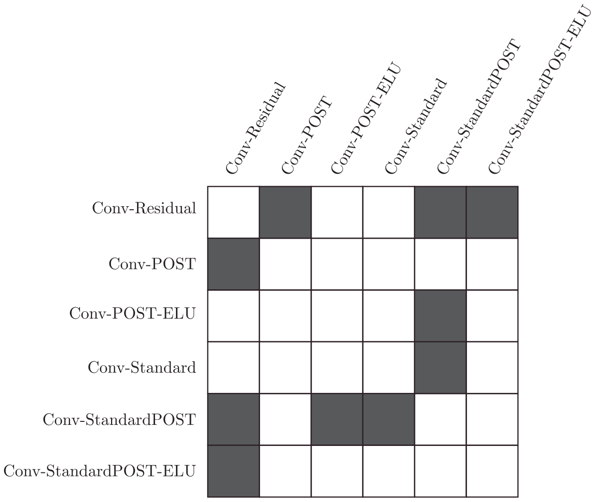
FIGURE 3. **Confusion matrices for the generated models over the evaluation dataset.**

and "Park", but other classes like "Airport" or "Bus" were degraded.

Finally, when considering the performance of networks implementing SE blocks together, from a general perspective, it is noticed that classes like "Street, Pedestrian", "Park" or "Public square" are improved with respect to the conventional residual network. Only the class "Airport" shows the best performance in the conventional network, followed by "Bus". The remaining classes are improved or worsened across all configurations in a degree not as significant as the aforementioned ones.

C. SIGNIFICANCE TEST

To determine if there are statistically significant differences in the performance of the different blocks analyzed in this work, a McNemar's test has been carried out [36]. This test, which is a paired non-parametric hypothesis test, has been widely recommended for evaluating deep learning models, which are often trained on very large datasets. The test is based on a contingency table created from the results obtained for two methods trained on exactly the same training test and evaluated on the same test set. The null hypothesis of the test is that the performance of the two analyzed systems disagree to the same amount. If the null hypothesis is rejected, there is evidence to suggest that the two systems have different performance when trained on a particular training set. Given a significance level α, if $p < \alpha$, there may be sufficient evidence to claim that the two classifiers show different proportions of errors. The result of applying the McNemar's test to all the available system pairs is shown in Fig. 4. Gray cells indicate p-values below a significance level of



	Conv-Residual	Conv-POST	Conv-POST-ELU	Conv-Standard	Conv-StandardPOST	Conv-StandardPOST-ELU
Conv-Residual		■			■	■
Conv-POST	■					
Conv-POST-ELU					■	
Conv-Standard					■	
Conv-StandardPOST	■		■	■		
Conv-StandardPOST-ELU	■					

$p < 0.05$

FIGURE 4. **Pairwise analysis of the studied residual networks using McNemar's test. Gray cells indicate *p*-values below a 0.05 significance level.**

0.05. It is confirmed that the two proposed blocks, *Conv-StandardPOST* and *Conv-StandardPOST-ELU*, show significant differences in performance with respect to all the other blocks but *Conv-POST*, which was the third best performing block. However, no significant differences can be observed between these new blocks, which only differ in the final ELU activation.

VI. CONCLUSION

The use of squeeze-excitation blocks in convolutional neural networks allows to perform a spatial and channel-wise recalibration of its inner feature maps. This work presented the use of squeeze-excitation residual networks for addressing the acoustic scene classification problem, and presented two novel block configurations that consider residual learning of standard and recalibrated outputs jointly. Results over the well-known DCASE dataset confirm that the proposed blocks provide meaningful improvements by adding a slight architecture modification, outperforming other competing approaches when no data augmentation or model ensembles are considered.

REFERENCES

[1] M. Valenti, S. Squartini, A. Diment, G. Parascandolo, and T. Virtanen, "A convolutional neural network approach for acoustic scene classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1547–1554.

[2] S. H. Bae, I. Choi, and N. S. Kim, "Acoustic scene classification using parallel combination of LSTM and CNN," in *Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE)*, 2016, pp. 11–15.

[3] L. Pham, H. Phan, T. Nguyen, R. Palaniappan, A. Mertins, and I. McLoughlin, "Robust acoustic scene classification using a multi-spectrogram encoder-decoder framework," 2020, *arXiv:2002.04502*. [Online]. Available: http://arxiv.org/abs/2002.04502

[4] Y. Han and K. Lee, "Acoustic scene classification using convolutional neural network and multiple-width frequency-delta data augmentation," 2016, *arXiv:1607.02383*. [Online]. Available: http://arxiv.org/abs/1607.02383

[5] I. Martín-Morató, M. Cobos, and F. J. Ferri, "A case study on feature sensitivity for audio event classification using support vector machines," in *Proc. IEEE 26th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Sep. 2016, pp. 1–6.

[6] I. Martín-Morató, M. Cobos, and F. J. Ferri, "Adaptive mid-term representations for robust audio event classification," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 12, pp. 2381–2392, Dec. 2018.

[7] G. Roma, W. Nogueira, and P. Herrera, "Recurrence quantification analysis features for auditory scene classification," Tech. Rep., 2013. [Online]. Available: http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/abstracts/SC/RNH.pdf

[8] D. Li, J. Tam, and D. Toub, "Auditory scene classification using machine learning techniques," Tech. Rep., 2013. [Online]. Available: http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/abstracts/SC/LTT.pdf

[9] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time-frequency representations for audio scene classification," Tech. Rep., 2013. [Online]. Available: http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/abstracts/SC/RG.pdf

[10] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification in DCASE 2019 challenge: Closed and open set classification and data mismatch setups," in *Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE)*. New York, NY, USA: New York Univ., Oct. 2019, pp. 164–168. [Online]. Available: http://dcase.community/documents/workshop2019/proceedings/DCASE2019Workshop_Mesaros_14.pdf

[11] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*. [Online]. Available: http://arxiv.org/abs/1710.09412

[12] M. D. McDonnell and W. Gao, "Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2020, pp. 141–145.

[13] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel 'squeeze & excitation'in fully convolutional networks," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2018, pp. 421–429.

[14] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[15] L. Yang, X. Chen, L. Tao, and X. Gu, "Multi-scale fusion and channel weighted CNN for acoustic scene classification," in *Proc. 2nd Int. Conf. Signal Process. Mach. Learn.*, Nov. 2019, pp. 41–45.

[16] J. Lee, T. Kim, J. Park, and J. Nam, "Raw waveform-based audio classification using sample-level CNN architectures," 2017, *arXiv:1712.00866*. [Online]. Available: http://arxiv.org/abs/1712.00866

[17] O. Akiyama and J. Sato, "Multitask learning and semisupervised learning with noisy data for audio tagging," DCASE2019 Challenge, Tech. Rep., 2019. [Online]. Available: https://pdfs.semanticscholar.org/95d3/1eb466b591161c7fd6fd8e14c146a3ccab71.pdf

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[20] A. Shah, E. Kadam, H. Shah, S. Shinde, and S. Shingade, "Deep residual networks with exponential linear unit," in *Proc. 3rd Int. Symp. Comput. Vis. Internet*, 2016, pp. 59–65.

[21] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," 2015, *arXiv:1511.07289*. [Online]. Available: http://arxiv.org/abs/1511.07289

[22] S. Perez-Castanos, J. Naranjo-Alcazar, P. Zuccarello, M. Cobos, and F. J. Ferri, "CNN depth analysis with different channel inputs for acoustic scene classification," 2019, *arXiv:1906.04591*. [Online]. Available: http://arxiv.org/abs/1906.04591

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] D. Fitzgerald, "Harmonic/percussive separation using median filtering," in *Proc. DAFX*, 2010, vol. 10, no. 4, pp. 1–4.

[25] J. Driedger, M. Müller, and S. Disch, "Extending harmonic-percussive separation of audio signals," in *Proc. ISMIR*, 2014, pp. 611–616.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[27] M. Wan, R. Wang, B. Wang, J. Bai, C. Chen, Z. Fu, J. Chen, X. Zhang, and S. Rahardja, "Ciaic-ASC system for DCASE 2019 challenge task1," DCASE2019 Challenge, Tech. Rep., 2019. [Online]. Available: http://dcase.community/documents/challenge2019/technical_reports/DCASE2019_Mou_41_t1.pdf

[28] F. Arabnezhad and B. Nasersharif, "Urban acoustic scene classification using binaural wavelet scattering and random subspace discrimination method," DCASE2019 Challenge, Tech. Rep., Jun. 2019. [Online]. Available: http://dcase.community/documents/challenge2019/technical_reports/DCASE2019_Fmta91_67.pdf

[29] S. Ma and W. Liu, "Acoustic scene classification based on binaural deep scattering spectra with neural network," DCASE2019 Challenge, Tech. Rep., Jun. 2019. [Online]. Available: http://dcase.community/documents/challenge2019/technical_reports/DCASE2019_MaLiu_112.pdf

[30] B. Ding, G. Liu, and J. Liang, "Acoustic scene classification based on ensemble system," DCASE2019 Challenge, Tech. Rep., Jun. 2019. [Online]. Available: http://dcase.community/documents/challenge2019/technical_reports/DCASE2019_DSPLAB_44.pdf

[31] Q. Kong, Y. Cao, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, "Cross-task learning for audio tagging, sound event detection and spatial localization: DCASE 2019 baseline systems," 2019, *arXiv:1904.03476*. [Online]. Available: https://arxiv.org/abs/1904.03476

[32] H. Liang and Y. Ma, "Acoustic scene classification using attention-based convolutional neural network," DCASE2019 Challenge, Tech. Rep., Jun. 2019. [Online]. Available: http://dcase.community/documents/challenge2019/technical_reports/DCASE2019_Liang_3.pdf

[33] D. Salvati, C. Drioli, and G. L. Foresti, "Urban acoustic scene classification using raw waveform convolutional neural networks," DCASE2019 Challenge, Tech. Rep., Jun. 2019. [Online]. Available: http://dcase.community/documents/challenge2019/technical_reports/DCASE2019_Salvati_35.pdf

[34] J. Naranjo-Alcazar, S. Perez-Castanos, I. Martin-Morato, P. Zuccarello, and M. Cobos, "On the performance of residual block design alternatives in convolutional neural networks for end-to-end audio classification," 2019, *arXiv:1906.10891*. [Online]. Available: http://arxiv.org/abs/1906.10891

[35] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 6964–6968.

[36] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998.

JAVIER NARANJO-ALCAZAR (Graduate Student Member, IEEE) received the Telecommunications degree and the master's degree in telecommunications engineering from the Universitat Politècnica de València, Valencia, Spain, in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Universitat de València, funded by the Torres Quevedo Program and the Valencian Start-Up Visualfy. His research interests include machine listening, few-shot learning, and open-set recognition. He was a recipient of the Best M.Sc. Thesis Award from the Regional Telecommunications Engineering Association, in 2019.

SERGI PEREZ-CASTANOS received the Telecommunications degree and the master's degree in telecommunications engineering from the Universitat Politècnica de València, Valencia, Spain, in 2016 and 2018, respectively. He is currently working as a Machine Learning Engineer with Visualfy. His research interests include machine listening, anomaly detection, and audio captioning.

PEDRO ZUCCARELLO received the Electronics Engineering degree from the University of Buenos Aires, Argentina, the M.Sc. degree in telecommunications from the Universitat Politècnica de Valencia, Valencia, Spain, and the Ph.D. degree from the Universitat de València, Valencia. He developed most of his career as a Researcher in several public Research and Development institutions, such as the Institute of Microelectronics of Barcelona, Barcelona, Spain, or the Institute of Corpuscular Physics, Valencia. Since 2017, he has been the Head of the Artificial Intelligence Group, Visualfy, private startup company. He received several Postdoctoral Fellowships such as Val-I+D, from the Valencian Government, or Torres Quevedo, from the Spanish Ministry of Science and Education. His research interests include artificial intelligence, machine learning, signal processing, electronics, and microelectronic design. He has coauthored nearly 30 articles in international peer-review journals and conferences in these areas.

MAXIMO COBOS (Senior Member, IEEE) received the master's degree in telecommunications and the Ph.D. degree in telecommunications engineering from the Universitat Politècnica de València, Valencia, Spain, in 2007 and 2009, respectively. In 2010, he received the Campus de Excelencia Postdoctoral Fellowship to work at the Institute of Telecommunications and Multimedia Applications. In 2011, he joined the Universitat de València, where he is currently an Associate Professor. His research interests include digital signal processing and machine learning for audio and multimedia applications. He has authored/coauthored more than 100 technical articles in international journals and conferences in these areas. He is a member of the Audio Signal Processing Technical Committee of the European Acoustics Association. He completed his studies with honors under the University Faculty Training Program (FPU) and was a recipient of the Ericsson Best Ph.D. Thesis Award from the Spanish National Telecommunications Engineering Association. He serves as an Associate Editor for the IEEE Signal Processing Letters.